**Práctica 1 Método Científico**

**Ejercicio 2. Dado el siguiente Código fuente:**

**public class Goldbach {**

**/\*\***

**\* Checks if the number n is prime. Negative numbers and zero are declared**

**\* to be non-prime.**

**\***

**\* @param n**

**\* Number to be tested.**

**\* @return True if n is prime, false if not.**

**\*/**

**public static boolean isPrime(Integer n) {**

**for (int p = 2; p < Math.sqrt(n)-1; p++) {**

**if (n % p == 0) {**

**return false;**

**}**

**}**

**return true;**

**}**

**/\*\***

**\* Checks Goldbach conjecture for n, that is, checks if the even number n**

**\* can be written as the sum of two prime numbers.**

**\***

**\* If n is odd, it returs automatically true.**

**\***

**\* @param n**

**\* Number to be checked.**

**\* @return True if n can be decomposed as sum of prime numbers, false if**

**\* not. Also prints the decomposition if true.**

**\*/**

**public static boolean checkGoldbach(Integer n) {**

**for (int p = 1; p < n / 2; p++) {**

**if (isPrime(p) && isPrime(n - p)) {**

**System.out.println(n + "=" + p + " & " + (n - p));**

**return true;**

**}**

**}**

**return false;**

**}**

**}**

**Solución**

|  |  |
| --- | --- |
| **Hipótesis** | El código funciona correctamente |
| **Predicción** | Al ejecutarlo la salida será correcta. |
| **Experimento** | Ejecutamos con los valores de entrada 13 y 23. |
| **Observación** | El resultado será 2 y 11 para el primer valor (correcto) y 4 y 19 (incorrecto, ya que 4 no es un primo) para el segundo. El programa no funciona correctamente. |
| **Conclusión** | Hipótesis rechazada. |

|  |  |
| --- | --- |
| **Hipótesis** | El programa no funciona para algunos valores. |
| **Predicción** | Al ejecutarlo la salida será incorrecta. |
| **Experimento** | Ejecutamos con los valores de entrada 2 y 3. |
| **Observación** | El programa no puede descomponerlos como una suma de primos. |
| **Conclusión** | Hipótesis aceptada. |

|  |  |
| --- | --- |
| **Hipótesis** | Dentro del bucle for de la función Goldbach cambiamos la condición de salida añadiendo un =. |
| **Predicción** | Al ejecutar el programa, éste funcionará correctamente. |
| **Experimento** | Ejecutamos con los valores de entrada 2 y 23. |
| **Observación** | El programa devuelve 1 y 1 (correcto) y 4 y 19 (incorrecto). No funciona correctamente. |
| **Conclusión** | Hipótesis rechazada. |

|  |  |
| --- | --- |
| **Hipótesis** | El bucle for de la función isPrime no contempla todos los casos y es incorrecto. |
| **Predicción** | Al ejecutar el código la salida será incorrecta. |
| **Experimento** | Ejecutamos con el valor de entrada 23. |
| **Observación** | El resultado es 4 y 19. |
| **Conclusión** | Hipótesis aceptada. |

|  |  |
| --- | --- |
| **Hipótesis** | Añadimos una condición que compruebe si los números son divisibles entre 2 y para el caso en el que el número de entrada sea 2 fuera del bucle for. |
| **Predicción** | Al ejecutar el código la salida será correcta. |
| **Experimento** | Ejecutamos el código con los valores de entrada 2 y 23. |
| **Observación** | El programa devuelve 1 y 1 (correcto). Sobre el 23 no devuelve nada ya que no puede descomponerse como una suma de primos. |
| **Conclusión** | Hipótesis aceptada y ejercicio resuelto. |